a) $T(n) = 3T(n-1) - 2T(n-2)$

$T(2) = 3T(1) - 2T(0)$
$T(3) = 3T(2) - 2T(1) = 7T(1) - 6T(0)$
$T(4) = 3T(3) - 2T(2) = 21T(1) - 18T(0) - 6T(1) + 4T(0)$
$= 15T(1) - 14T(0)$
$T(5) = 3T(4) - 2T(3) = 45T(1) - 42T(0) - 14T(1) + 12T(0)$
$= 31T(1) - 30T(0)$

$T(2) = 3T(1) - 2T(0)$
$T(3) = 7T(1) - 6T(0)$
$T(4) = 15T(1) - 14T(0)$
$T(5) = 31T(1) - 30T(0)$ so,
$T(n) = (2^n - 1)\,T(1) - (2^n - 2)\,T(0)$

b) $T(n/2) + 1$

$a = 1$, $b = 2$ $f(n) = 1$ $d = 1$

$a = b^d = n^0 = 1$, $f(n) \in O(1)$

$T(n) \in \Theta(n^0 \log n) = \Theta(\log n)$.

d) $T(n) = 4T(n/2) + n^2$

$aT(\frac{n}{b}) + f(n)$,

$a = 4$ $\quad\quad a = b^d$ since $4 = 2^2$

$b = 2$

$f(n) = n^2 \in \Theta(n^2)$, $d = 2$

so, $T(n) \in \Theta(n^2 \log n)$. $(a = b^d)$

e) $T(n) = 2T(n/2) + O(n)$

$aT(\frac{n}{b}) + f(n)$

$a = 2$ $\quad\quad 2 = 2^1 \; (a = b^d)$

$b = 2$.

$f(n) \in O(n^1)$, $d = 1$

so, $T(n) \in \Theta(n^1 \log n)$

f) $T(n) = T(n/2) + T(n/4) + n$

$= T(n/4) + T(n/8) + n/2 + T(n/4) + n$

$= 2T(n/4) + T(n/8) + 3n/2$

$= 2(T(n/8) + T(n/16) + (n/4)) + T(n/8) + 3n/2$

$= 3T(n/8) + T(n/16) + 4n/2$

$\hookrightarrow$ $T(n) = T(n/2) + T(n/4) + n$

$= 2T(n/4) + T(n/8) + 3n/2$

$= 3T(n/8) + T(n/16) + 4n/2$

$\swarrow$ (after k times)

$T(n) = kT(n/2^k) + T(n/2^{k+1}) + (k+1)n/2$

if $n/2^{k+1} = 1$, $k = \log_2(n/2)$

$T(n) = \log_2(n/2)\, T(2) + T(1) + (\log_2(n/2) + 1)n/2$

so $T(n) \in \Theta(n \log n)$.

Scanned with CamScanner

g) $T(n) = T(n/2) + n$

$a = 1,\ b = 2,\ f(n) = n \in O(n),\ d = 1$

$a < b^d$ so $T(n) \in \Theta(n)$.

h) $T(n) = 2T(\sqrt{n}) + 1$

if $S(m) = T(2^m)$,

$S(m) = 2S(m/2) + 1$

$a = 2,\ b = 2,\ f(m) = 1,\ d = 0$

$a > b^d,\quad S(m) \in \Theta(n^{\log_2 2}) = \Theta(n)$

3)

a)

$T(n) = 5T(n/2) + n^3$

$a = 5,\ b = 2\quad f(n) = n^3 \in O(n^3),\ d = 3$

$a < b^d,\quad T(n) \in \Theta(n^3)$

b) $T(n) = 2T(n-2) + n$

$T(2) = 2T(0) + 2 \qquad 2 \times 1$ $\downarrow 3$

$T(4) = 4T(0) + 8 \qquad 2 \times 4$ $\downarrow 7$

$T(6) = 8T(0) + 22 \qquad 2 \times 11$ $\downarrow 15$

$T(8) = 16T(0) + 52 \qquad 2 \times 26$

$\Big\rangle$ 2^n sums: $2(2^n - 1)$

-1 for each

$\uparrow$

$T(n) = 2^{n/2}T(0) + 2(2(2^{n/2} - 1) - n/2)$

from this equation $T(n) \in O(2^n)$

Scanned with CamScanner

c)

$T(n) = 3T(n/2) + n^2$

$a=3,\ b=2,\ f(n)=n^2 \in O(n^2),\ d=2$

$3 < 2^2,\ a < b^d$ so $T(n) \in \Theta(n^2)$

I would choose the third algorithm because in terms of big-O, it is the fastest among three of them.

5)

$T(n) = \underset{\text{from recursion.}}{2T(n/2)} + \underset{\text{from loop}}{n}$

$T(n) = 4T(n/4) + 3n = 8T(n/8) + 7n$

k times

$T(n) = 2^k T(n/2^k) + (2^k - 1)n$

$T(2^k) = 2^k T(1) + \underbrace{2^{2k} - 2^k}$

when $n = 2^k$,

this numbers of "a" will be printed.

2)

height of tree:

$T(n) = 2T(n/2) + 2 \rightarrow$ 2 if checks.

$2T(n/2) \rightarrow$ node left and right.

$a = 2$, $b = 2$, $f(n) = 2 \in O(1)$, $d = 0$

$a > b^d$, $T(n) \in \Theta(n^{\log_2 2}) = \Theta(n)$

is_balanced:

$T(n) = 2S(n/2) + 2T(n/2)$

$2S(n/2) \rightarrow$ height of tree.

$T(n) = 2T(n/2) + O(n)$.

$a = 2$, $b = 2$, $d = 1$

$a = b^d$, $T(n) \in \Theta(n \log n)$

Scanned with CamScanner

4) In best case, all the elements in the graph are 0 and we just check all of them in $O(M \times N)$

Scanned with CamScanner